

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
11 August 2011 (11.08.2011)

(10) International Publication Number
WO 2011/097057 A2

- (51) **International Patent Classification:**
G06F 17/30 (2006.01)
- (21) **International Application Number:**
PCT/US2011/021210
- (22) **International Filing Date:**
13 January 2011 (13.01.2011)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
12/700,985 5 February 2010 (05.02.2010) US
- (71) **Applicant (for all designated States except US):** MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (72) **Inventors:** SRIKANTH, Munirathnam; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). VADDADI, Phani; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). NAJM, Tarek; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) **Designated States (unless otherwise indicated, for every kind of national protection available):** AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States (unless otherwise indicated, for every kind of regional protection available):** ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

[Continued on next page]

(54) **Title:** CONTEXTUAL QUERIES

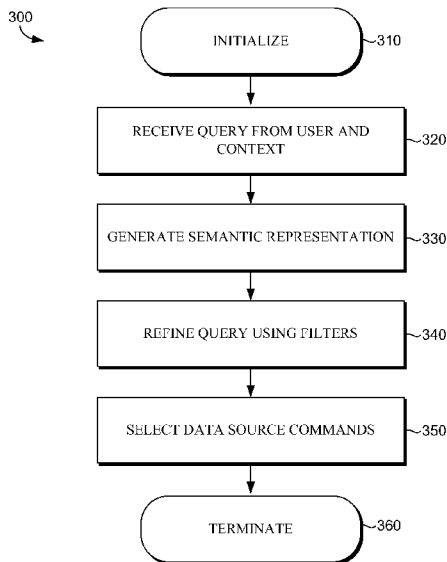


FIG. 3

(57) **Abstract:** Systems, methods, and computer-storage media for generating contextual queries are provided. The system includes a search engine to receive user queries and contexts, a query understanding component to generate a semantic representation of the query, and a data source command generator to transform the semantic representation into commands for multiple data sources. The data source command generator is connected to the query understanding component. The data source command generator selects data source commands based on lexical information associated with each data source.



WO 2011/097057 A2

- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*
- Published:**
 - *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

CONTEXTUAL QUERIES

BACKGROUND

[0001] Conventional search engines receive queries from users to locate web pages having terms that match the terms included in the received queries. Conventionally, the search engines ignore the context and meaning of the user query and treat the query as a set of words. The terms included in the query are searched for based on frequency, and results that include the terms of the query are returned by the search engine.

[0002] Accordingly, conventional search engines return results that might fail to satisfy the interests of the user. The user attempts to reformulate the query by choosing words that are likely found in a document of interest. For instance, a user looking for stock information may enter a query for “PE Company A Stock.” The conventional search engine will treat each word separately and return documents having the term “Company A,” documents having the term “PE,” documents having the terms “stock,” and documents having any of the terms. The conventional search engine is unable to intelligently select documents in results that discuss the stock performance of Company A, a comparison of Company A to its competitor, and news about the management of Company A. The user must read the different documents in the results to determine whether any of the documents includes performance information.

SUMMARY

[0003] Embodiments of the invention relate to systems, methods, and computer-readable media for generating contextual queries. A search engine receives a query from a client device along with context information provided by applications utilized during the current search session. In turn, a query understanding component processes the context information and query to generate a semantic representation of the query. The semantic

representation of the query is further processed by a data source command generator to select several data source commands based on lexical information associated with each data source. The data source commands are issued from the data source command generator to the data sources to return answers and results to the search engine in response to the user query and generated contextual queries.

[0004] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the detailed description. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in isolation to determine the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] Illustrative embodiments of the invention are described in detail below with reference to the attached drawing figures, which are incorporated by reference herein, wherein:

[0006] FIG. 1 is a block diagram illustrating an exemplary computing device in accordance with embodiments of the invention;

[0007] FIG. 2 is a network diagram illustrating exemplary components of a computer system configured to generate contextual queries in accordance with embodiments of the invention; and

[0008] FIG. 3 is a logic diagram illustrating a computer-implemented method for generating contextual queries in accordance with embodiments of the invention.

DETAILED DESCRIPTION

[0009] This patent describes the subject matter for patenting with specificity to satisfy statutory requirements. However, the description itself is not intended to limit the scope of this patent. Rather, the inventors have contemplated that the claimed subject matter might also be embodied in other ways, to include different steps or combinations of steps similar to the ones described in this patent, in conjunction with other present or future technologies. Moreover, although the terms “step” and “block” may be used herein to connote different elements of methods employed, the terms should not be interpreted as implying any particular order among or between various steps herein disclosed unless and except when the order of individual steps is explicitly described.

[0010] As used herein the term “contextual query” refers to data commands that are used to access data sources and to retrieve information from the data sources based on the structure of the information included in the data source and the meaning of terms included in a query.

[0011] As used herein the term “component” refers to any combination of hardware, firmware, and software.

[0012] Embodiments of the invention provide contextual queries that allow a user to receive answers to the user query. The answers are collected from a large collection of content having structured data, semistructured data, and unstructured data. The contextual queries are generated by a computer system based on ontologies associated with the terms included in the query and applications that are utilized during a current search session. The contextual queries search a combination of structured, unstructured, and semistructured data for answers to the query.

[0013] For instance, answers for a user's finance queries are returned by a computer system using an ontology-driven understanding of the finance queries. The computer system may provide a finance application that enables users to obtain answers to natural language queries like "Companies with similar PE ratio as Company A," "Company A operating income," and "Company A and B valuation." An ontology is used by the computer system to understand a number of key ratios that should be calculated using current data in response to the queries. The key ratios may include, among other things, "PE Ratio," "Price-to-Sales ratio," etc. The natural language queries and corresponding ontologies are used to generate a semantic representation. The finance application may transform the semantic representation to a SPARQL query that is issued to data sources, which include records that are traversed to return the answers. In turn, the finance application returns a comparison of these ratios and other valuation metrics for the companies A and B.

[0014] The computer system receives queries from a user. Also, the computer system receives contexts for one or more applications traversed during the current search session. Using one or more ontologies, the computer system understands the queries and contexts and generates semantic representations of the queries and the contexts of the applications in which a user formulates the query. Concepts, instances, properties, and relations are included in the semantic representation of the queries based on the taxonomy, templates, or definitions included in the ontologies. In one embodiment, the semantic representation is a graph of the query using the extracted concepts, instances, properties, and relations, etc. In turn, the semantic representation may be automatically transformed by the data source command generator into data source specific query languages to retrieve relevant information and answers to the query.

[0015] FIG. 1 is a block diagram illustrating an exemplary computing device 100 in accordance with embodiments of the invention. The computing device 100 includes bus 110, memory 112, processors 114, presentation components 116, input/output (I/O) ports 118, input/output (I/O) components 120, and a power supply 122. The computing device 100 is but one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing device 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated.

[0016] The computing device 100 typically includes a variety of computer-readable media. By way of example, and not limitation, computer-readable media may comprise Random Access Memory (RAM); Read Only Memory (ROM); Electronically Erasable Programmable Read Only Memory (EEPROM); flash memory or other memory technologies; CDROM, digital versatile disks (DVD) or other optical or holographic media; magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium that may be used to encode desired information and be accessed by the computing device 100. Embodiments of the invention may be implemented using computer code or machine-useable instructions, including computer-executable instructions such as program modules, being executed by a computing device 100, such as a personal data assistant or other handheld device. Generally, program modules including routines, programs, objects, modules, data structures, and the like, refer to code that performs particular tasks or implements particular abstract data types. Embodiments of the invention may be practiced in a variety of system configurations, including distributed computing environments where tasks are performed by remote-processing devices that are linked through a communications network.

[0017] The computing device 100 includes a bus 110 that directly or indirectly couples the following components: a memory 112, one or more processors 114, one or more presentation components 116, input/output (I/O) ports 118, I/O components 120, and an illustrative power supply 122. The bus 110 represents what may be one or more busses (such as an address bus, data bus, or combination thereof). Although the various components of FIG. 1 are shown with lines for the sake of clarity, in reality, delineating various modules is not so clear, and metaphorically, the lines would more accurately be grey and fuzzy. For example, one may consider a presentation component 116 such as a display device to be an I/O component 120. Also, processors 114 have memory 112. Distinction is not made between “workstation,” “server,” “laptop,” “handheld device,” etc., as all are contemplated within the scope of FIG. 1.

[0018] The memory 112 includes computer-readable media and computer-storage media in the form of volatile and/or nonvolatile memory. The memory may be removable, nonremovable, or a combination thereof. Exemplary hardware devices include solid-state memory, hard drives, optical-disc drives, etc. The computing device 100 includes one or more processors 114 that read data from various entities such as the memory 112 or I/O components 120. The presentation components 116 present data indications to a user or other device. Exemplary presentation components 116 include a display device, speaker, printer, vibrating module, and the like. The I/O ports 118 allow the computing device 100 to be physically and logically coupled to other devices including the I/O components 120, some of which may be built in. Illustrative I/O components 120 include a microphone, joystick, game pad, satellite dish, scanner, printer, wireless device, and the like.

[0019] A computer system that generates contextual queries includes a search engine, a query understanding component, and a data source command generator. The contextual queries are generated based on ontologies associated with a query provided by

the client device and ontologies corresponding to applications that formulated the query. The search engine receives the query from the client device and context from the application. In turn, the search engine transmits answers to the contextual queries and results that include content corresponding to the query to a client device. The client device displays the results along with the answers.

[0020] FIG. 2 is a network diagram illustrating exemplary components of a computer system configured to generate contextual queries in accordance with embodiments of the invention. The computer system 200 has a client device 210, a network 220, search engine 230, data source command generator 240, and query understanding component 250.

[0021] The client device 210 is connected to the search engine 230 via network 220. In some embodiments, the client device 210 may be any computing device that is capable of web accessibility. As such, the client device 210 might take on a variety of forms, such as a personal computer (PC), a laptop computer, a mobile phone, a personal digital assistant (PDA), a server, a CD player, an MP3 player, a video player, a handheld communications device, a workstation, any combination of these delineated devices, or any other device that is capable of web accessibility.

[0022] The client device 210 allows a user to enter queries. The client device 210 transmits the queries to the search engine 230. In certain embodiments, the client device 210 also transmits a context associated with an application used by the user to formulate the query. In other embodiments, the search engine 230 may provide the context. In turn, the client device 210 receives results that include answers. The client device 210 may also display the answers for the contextual queries and results for the queries to the users.

[0023] The network 220 connects the client device 210, search engine 230, data source command generator 240, and query understanding component 250. The network 220 may be wired, wireless, or both. The network 220 may include multiple networks, or a network of networks. For example, the network 220 may include one or more wide area networks (WANs), one or more local area networks (LANs), one or more public networks, such as the Internet, or one or more private networks. In a wireless network, components such as a base station, a communications tower, or even access points (as well as other components) may provide wireless connectivity in some embodiments. Although single components are illustrated for the sake of clarity, one skilled in the art will appreciate that the network 220 may enable communication between any number of client devices 210.

[0024] The search engine 230 is a server computer that provides results for queries received from client devices 210 and answers to contextual queries dynamically generated by the data source command generator 240. The search engine 230 is configured to receive user queries and application contexts. The search engine 230 returns results to the user queries and answers to the contextual queries. In some embodiments, the search engine 230 returns only answers to the contextual queries.

[0025] The data source command generator 240 is communicatively connected to the query understanding component 250. In one embodiment, the data source command generator 240 includes filters 242 and lexical information 244. The data source command generator 240 is configured to transform a semantic representation provided by the query understanding component 250 into contextual queries that are applied across multiple data sources using data source commands selected based on lexical information 244 associated with each data source. In one embodiment, the data sources include unstructured, structured, or semistructured data sources that are queried using one of: Structured Query

Language (SQL), SPARQL Protocol and RDF Query Language (SPARQL), or textual keyword queries.

[0026] The filters 242 are configured to refine the semantic representation provided by the query understanding component 250. The filters 242 refine the semantic representation by augmenting the semantic representation based on information associated with a current query session. In one embodiment, the filter 242 includes domain, collaboration, geographic, temporal, task, and user. The domain filter may be identified based on application context. For instance, a user searching for stock may be interested in the finance domain or cooking domain. The data source command generator 240 may select one of these domains based on the other filters 242. For instance, the collaboration or community filters may indicate that the user is connected to cooking groups or finance groups. Based on the group associated with the user, the data source command generator 240 selects the relevant domain filter. For instance, a user in a finance community is likely interested in company stock. Accordingly, the domain filter may be set to finance. The geographic filter may include the current location of the user. For example, a user that is a British subject in England may issue the query, and the geographic filter may set to England by the computer system 200. The temporal filter may include the time frame for the query. If a date or time is not included in the query, the temporal filter may be set to the current 24-hour time period. The task filter may be selected based on the application used by the user or it may be inferred by the computer system 200. The task filter may include checking email, searching for content, editing a document, etc. The user filter may include profile attributes that override values selected for the other filters 242. For instance, a British subject user may specify in a profile that interests include U.S. stock. Accordingly, the geographic information for the stock query may be changed from England to United States of America. These values for the filters 242 are included in the

semantic representation that is transformed to the contextual queries by the data source command generator 240.

[0027] The lexical information component 244 is a part of the data source command generator 240. The lexical information component 244 is used to select data sources that should receive the contextual queries. The data sources having lexical information similar to the terms of the contextual query are selected to receive the contextual query. In one embodiment, the lexical information component 244 stores lexical information associated with ontologies and the data sources that store the content that is traversed by the search engine. In some embodiments text mining is performed on each data source by the computer system to select lexical information associated with each data source that is stored in the lexical information component 244.

[0028] In turn, the data source command generator 240 generates contextual queries that are issued against the selected data sources. The semantic representation of the query may be further processed to select appropriate commands for the selected data sources. The lexical information component 244, ontologies 252, and rules 254 are accessed to identify methods for each data source that access or compute relevant information from data sources based on the semantic description of concepts, properties, and relations included in the semantic representation.

[0029] The query understanding component 250 is configured to store ontologies 252 that identify the query type and context type. The query understanding component 250 also provides rules 254 associated with items in the ontologies 252 that expand a semantic representation and identify methods that access or compute relevant information from data sources based on a semantic description of concepts, properties, and relations expressed in the ontologies 252. For instance, the query type may identify one or more

input types and one or more output types, where the input and output types correspond to concepts, instances, properties, or relations in a query ontology or a context ontology.

[0030] The query understanding component 250 is configured to generate the semantic representation of the query and context. The query ontology and context ontology may be used by the query understanding component 250 to generate a graph of the semantic representation based on the uniform resource identifiers (URIs) associated with the each concept, instance, etc., included in the ontologies 252. The semantic representation is conditioned by the appropriate context rules of the user query and application context.

[0031] In one embodiment, the graph is a semantic query representation (SQR) that is formatted in extensible markup language (XML). The SQR may include the following nodes: query type (QT), Context (CXT), and Display (DSP) templates. For instance, a natural language query for Company A may have the following SQR <QT: Instance Profile: Company; Instance Type: Stock; Instance URI: Company A> <CXT: Natural Language, Finance> <DSP: Company Name, Symbol, News>.

[0032] Depending on the data source, the semantic query representation is translated into data source-specific queries to retrieve relevant answers and information from those data sources. In the semantic representation, QT is a formal representation of a hierarchy of the different types of queries an application can expect from its users. Each QT is qualified by a set of input and output parameters that facilitate extensions to other parts of the ontology. CXT is a formal representation of a hierarchy of the different types of contexts in which the user query can be captured and interpreted. CXT can be explicitly identified by the applications a user interacts to issue their queries or implicitly derived from the query text. Each CXT identifies the conditions and criteria for interpreting concepts, instances, etc., in a given query. The CXT may either expand or

disambiguate the concepts, instances, etc. included in the semantic query representation. The semantic graph is structured according to QT and CXT templates associated with the query type in the query-type ontology and the context type in the context-type ontology and lexical information associated with each data source.

[0033] The ontologies 252 include words or phrases that correspond to content in the data sources. Each ontology 252 includes a taxonomy for a domain and the relationship between words or phrases in the domain. The domains may include medicine, art, computers, etc. In one embodiment, the ontologies 252 also store the query type and context type. The query type identifies the type and structure of textual user queries. For instance, the query type may include natural language, structured, in-line command, etc. The context type identifies and organizes the different types of contexts in which queries can be expressed. For instance, the context may include search engine, email application, finance application, etc. The ontology 252 is associated with rules 254. The rules may be referenced in the semantic query representation using a URI corresponding to the appropriate rule.

[0034] The rules 254 identify the concepts, instances, properties, and relations across a number of domains. In certain embodiments, the rules 254 may define methods or functions that are used to compute results from data included in the data sources. For instance, the rules 254 may include comparators, mathematical functions, statistical functions, or other heuristics. In other embodiments, the rules 254 may also be associated with functions identified in the commands available for the data sources.

[0035] In an embodiment, answers are transmitted to a client device based on contextual queries that are dynamically generated from queries received from a user of the client device. The computer system may execute a computer-implemented method for

dynamically generating contextual queries based on ontologies corresponding to the query and application traversed by the user when formulating the query.

[0036] FIG. 3 is a logic diagram illustrating a computer-implemented method for generating contextual queries in accordance with embodiments of the invention. The method initializes in step 310 when the computer system is connected to a network of client devices.

[0037] In step 320, the computer system receives a user query and application context. The context is provided by an application that the user operates during the current query session. In step 330, the computer system generates a semantic representation of the query using domain ontologies. The computer system determines a query and context type using query and context ontologies. The structure for the semantic representation is derived from a template associated with a query type included in the query ontology or context type included in the context ontology.

[0038] The semantic representation is refined using filters associated with each domain traversed in a current query session, in step 340. The filters augment the semantic representation with criteria associated with one or more applications operated by the user during the current query session request. In step 350, the computer system selects one or more data source commands to issue against data sources having content associated with terms in the query. In certain embodiments, the data sources include structured data sources, unstructured data sources, and semistructured data sources. The structured data sources are queried using one of: Structured Query Language (SQL) or SPARQL Protocol and RDF Query Language (SPARQL). The unstructured or semistructured data sources are queried using textual keyword queries. The method terminates in step 360.

[0039] In summary, contextual queries allow a user to traverse multiple data sources using information built into ontologies associated with a query provided by a user and ontologies associated with applications utilized to generate the query. For instance, a user may send a query to a search engine, which returns a number of results. In addition, the search engine may also provide answers associated with contextual queries formulated from the user query.

[0040] For instance, answers for a user's finance queries are returned by the computer system using an ontology-driven understanding of the finance queries. The computer system may provide a finance application that enables users to obtain answers to natural language queries like "Companies with similar PE ratio as Company A," "Company A operating income," and "Company A and B valuation." The latter query processing may use the ontology to understand valuation includes a number of key ratios that should be calculated using current data. The key ratios may include, among other things, "PE Ratio," "Price-to-Sales ratio," etc. In turn, the finance application returns a comparison of these ratios and other valuation metrics for the companies A and B. The natural language queries and corresponding ontologies are used to generate a semantic representation. The finance application may transform the semantic representation to a SPARQL query that is issued to data sources, which include records that are traversed to return the answers.

[0041] The computer system receives a query: "MSFT higher PE" from a user and application context: "finance" and "natural language" from the finance application. The computer system identifies the finance ontology and natural language ontology. In turn, the query is transformed using a linguistic interpretation to "MSFT higher/high-1 PE." The finance ontology may be used to further transform the semantic representation using mathematical operators and the definition for PE ratio. The semantic representation is

transformed to “MSFT higher/high-1 Comparator PE.” The computer system may apply a number of filters to refine the semantic representation. In an embodiment, the semantic representation is a query graph generated using the ontologies. The filters include domain, collaboration, community, geographic, temporal, task, and user interests. After applying these filters, the computer system transforms the query to “Ticker MSFTUS/Ticker higher greater-1PE/PE ratio Date: today.” In turn the semantic representation is converted to data source commands. The computer system may issue the following: “Ticker.PeRatio>MSFTQUS.PERatio&&Date=today.” This data source command is sent to multiple data sources to locate the answers.

[0042] Many different arrangements of the various components depicted, as well as components not shown, are possible without departing from the spirit and scope of the present invention. Embodiments of the invention have been described with the intent to be illustrative rather than restrictive. It is understood that certain features and subcombinations are of utility and may be employed without reference to other features and subcombinations and are contemplated within the scope of the claims. Not all steps listed in the various figures need be carried out in the specific order described.

CLAIMS

The invention claimed is:

1. A computer-implemented method to generate contextual queries, the method comprising:

receiving a user query and context;

generating a semantic representation of the query using domain ontologies, wherein the domain ontologies identify filters, concepts, and relations in a number of categories;

refining the semantic representation using filters associated with each domain traversed in a current query session;

and

selecting one or more data source commands to issue against data sources having content associated with terms in the query based on the context and the semantic representation.

2. The method of claim 1, wherein the context is provided by an application that the user operates during the current query session.

3. The method of claim 1, further comprising: determining a query and context type using query and context ontologies, wherein structure for the semantic representation is derived from a template associated with a query type included in the query ontology and some filters augment the semantic representation with criteria that one or more applications operated by the user during the current query session request.

4. The method of claim 1, wherein the data source includes structured data sources, unstructured data sources, and semistructured data sources.

5. The method of claim 4, wherein the structured data sources are queried using one of: Structured Query Language (SQL) or SPARQL Protocol and RDF Query Language (SPARQL).

6. The method of claim 4, wherein the unstructured or semistructured data sources are queried using textual keyword queries.

7. The method of claim 4, wherein the data source commands are selected based on lexical information associated with each data source.

8. One or more computer readable media configured to perform a method to process queries, the method comprising:

receiving a user query and context;

generating a semantic representation of the query using domain ontologies;

refining the semantic representation using filters associated with each domain traversed in a current query session;

and

selecting one or more data source commands to issue against a data source having content associated with terms in the query.

9. The media of claim 8, wherein the context is provided by an application that the user operates during the current query session.

10. The media of claim 8, further comprising: determining a query and context type using query and context ontologies, wherein structure for the semantic representation is derived from a template associated with a query type included in the query ontology and the filters augment the semantic representation with criteria that one or more applications operated by the user during the current query session request.

11. The media of claim 8, wherein the data source includes structured data sources, unstructured data sources, and semistructured data sources.

12. The media of claim 11, wherein the structured data sources are queried using one of: Structured Query Language (SQL) or SPARQL Protocol and RDF Query Language (SPARQL).

13. The media of claim 11, wherein the unstructured or semistructured data sources are queried using textual keyword queries.

14. A computer system having processors and memories configured to generate contextual queries, the system further comprising:

a search engine configured to receive user queries and contexts;

a query understanding component configured to store ontologies that identify the query type and application type, the query type identifies one or more input types and one or more output types, wherein the input and output types correspond to concepts, instances, properties, or relations in a domain ontology or an application ontology;

and

a data source command generator communicatively connected to the query understanding component, wherein the data source command generator is configured to transform a semantic representation provided by

the query understanding component into contextual queries that are applied across multiple data sources using data source commands selected based on lexical information associated with each data source.

15. The computer system of claim 14, wherein the ontologies provide rules that expand the semantic query representation and identify methods that access or compute relevant information from data sources based on a semantic description of concepts, properties, and relations expressed in the ontologies; and the data sources include unstructured, structured, or semistructured data sources and are queried using one of: Structured Query Language (SQL), SPARQL Protocol and RDF Query Language (SPARQL), or textual keyword queries.

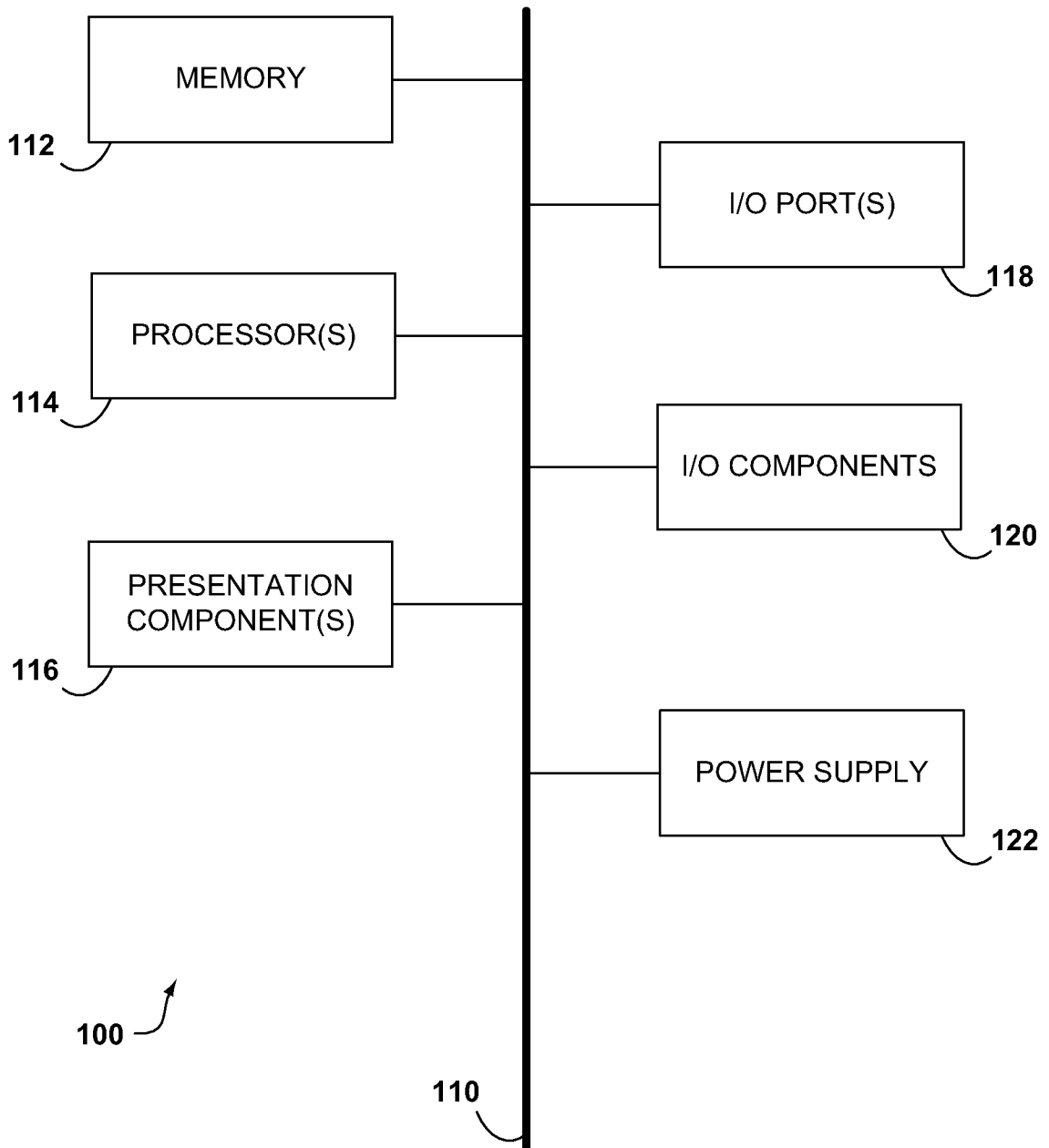


FIG. 1

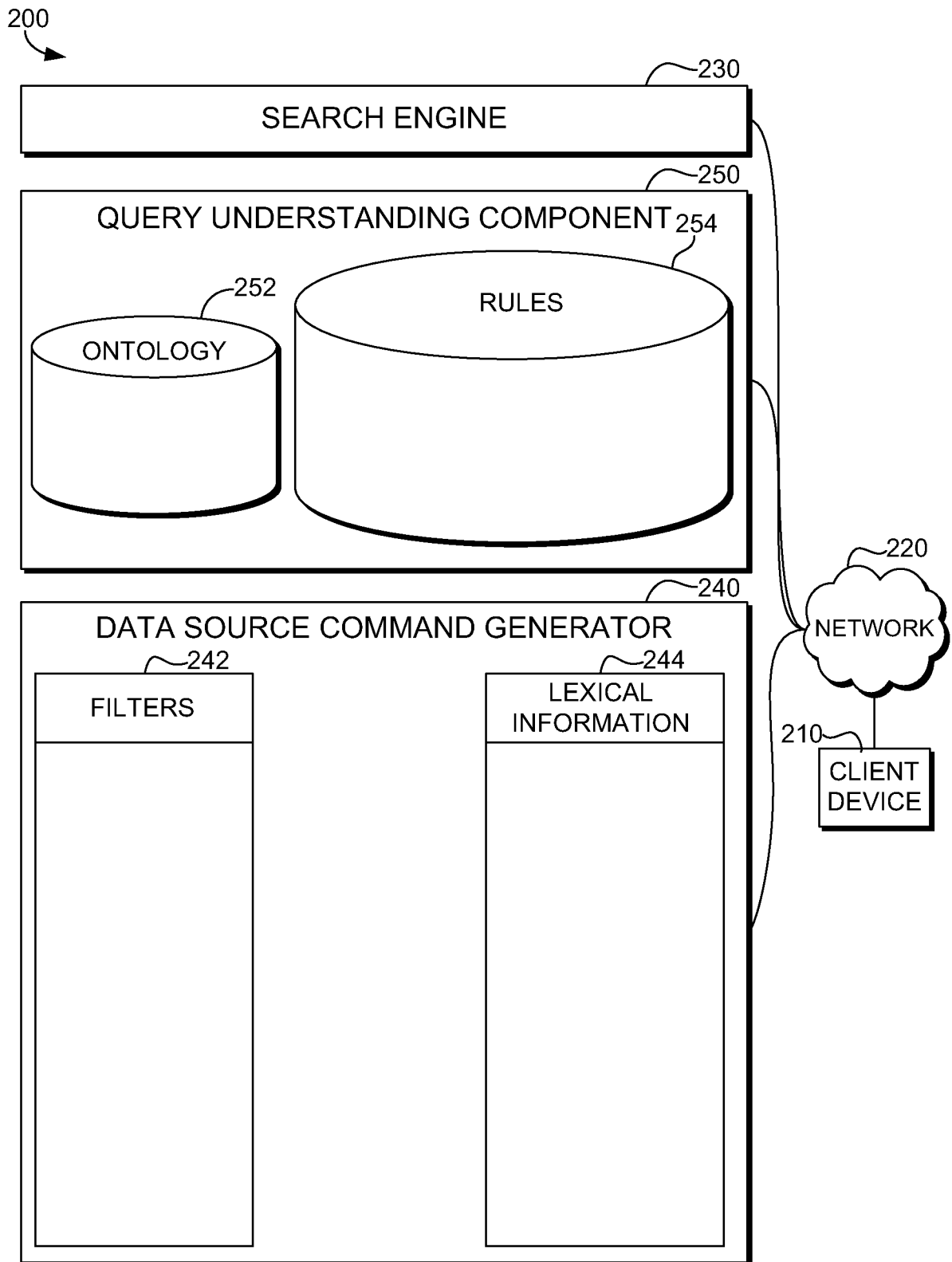


FIG. 2

3/3

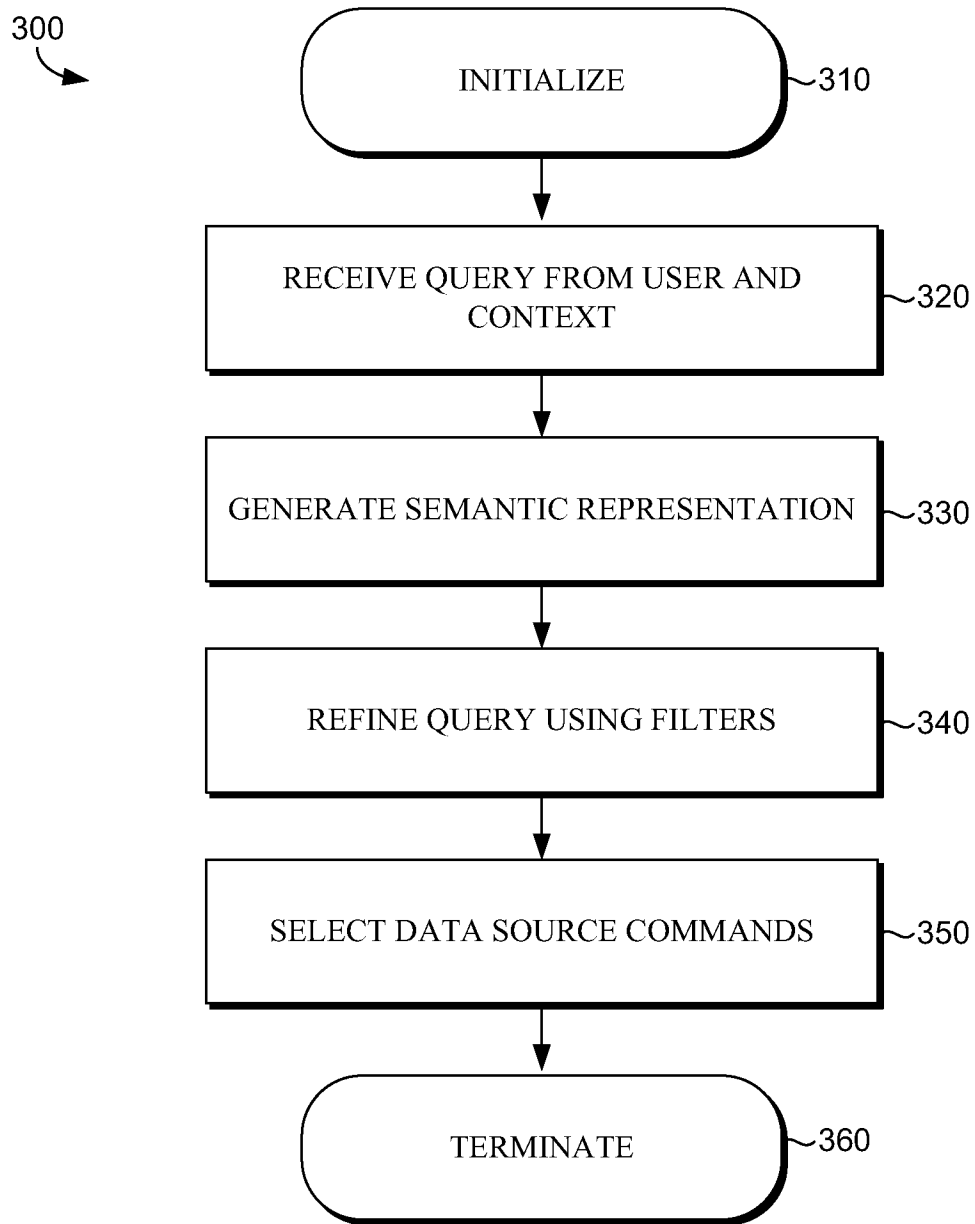


FIG. 3